A close up of a logo

Description automatically generated

**Disclaimer: The content is curated from online/offline resources and used for educational purpose only**

**Code Repository**

**LAB MANUAL**

**Managing Branches for Feature Development**

**Objective:**

* Create new branches for feature development.
* Switch between branches.
* Merge feature branches into the main branch.
* Understand best practices for branch management in team projects.

**Equipment Required:**

* Computer with internet access
* Git installed on the system
* GitHub or GitLab account (or any remote repository provider)
* Git Bash (Windows) or Terminal (macOS/Linux)
* Code/text editor (e.g., VS Code, Notepad++)

**Prerequisites:**

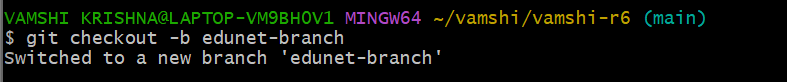
* Familiarity with Git basics (init, add, commit)
* Understanding of local and remote repositories
* A GitHub repository (remote origin) already set up

**Problem Statement:**

You are developing a new feature in a project hosted on GitHub. To avoid conflicts with other developers, you will create a new branch, make changes in that branch, and merge it back into the main branch once the feature is complete.

**Procedure:**

**Step 1: Create a New Branch**

****

**Step 2: Make Changes in the Feature Branch**

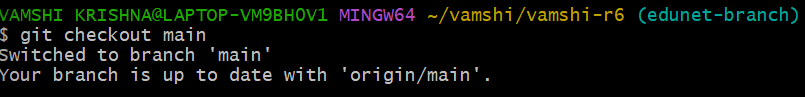
* Edit or create a file (e.g., feature.txt) in your editor.
* Add content like: This is a new feature being developed in a separate branch.
* Save the file.

**Step 3: Stage and Commit the Changes**

A screenshot of a computer

AI-generated content may be incorrect.

**Step 4: Switch Back to Main Branch**

****

**Step 5: Merge the Feature Branch into Main**

**A black screen with white text

AI-generated content may be incorrect.**

**Step 6: Push the Changes to GitHub**

**A computer screen shot of a computer code

AI-generated content may be incorrect.**